General Methodology in Two Dimensions for Classical Simulation of Reactive and Nonreactive Events on *Ab Initio* Potential Energy Surfaces

MICHAEL R. SALAZAR, RICHARD L. BELL

Henry Eyring Center for Theoretical Chemistry, Chemistry Department, University of Utah, Salt Lake City, Utah 84112

Received 19 November 1997; accepted 27 March 1998

ABSTRACT: A completely general two-dimensional (2D) methodology for the classical simulation of reactive and nonreactive events on *ab initio* potential energy surfaces is introduced and tested. The methodology requires the minimum amount of information given *a priori*—geometries and energies at these geometries. From a list of *ab initio* geometries and energies, simulations may be executed and a distribution of outcomes obtained. The method introduced attempts a local approach at simulating the dynamics of the system, rather than a global analytic fit to the potential energy surface. © 1998 John Wiley & Sons, Inc. J Comput Chem 19: 1431–1444, 1998

Keywords: *ab initio* potential energy surface; domain; tessellation; interpolation; barycentric coordinates

Introduction

The dynamics of molecular systems has received considerable interest during the past few decades. Often, theoreticians and investigators are not interested in the static properties of systems, but in how these properties change during the course of and following a chemical reaction. This interest has been aided by recent computational advances which has made the solution of the time-dependent and time-independent Schrödinger equation feasible.

Perhaps the largest difficulty in simulating the dynamics of a chemical reaction lies in the construction of the potential energy surface (PES). Often, for thermal reactions the Born–Oppenheimer approximation (BOA) is sufficient and the reaction may be thought of as proceeding on a single PES. For the simulation of reactions where the BOA is known to break down, the problem

Correspondence to: M. R. Salazar

Contract/grant sponsor: National Science Foundation; contract/grant number: CHE9618904

CCC 0192-8651 / 98 / 131431-14

becomes much more difficult. Methods have been developed for the simulation of these reactions, but the computational costs are great.[1] The methodology that forms the emphasis of this paper is valid where the BOA is valid. Therefore, we are interested in facilitating Born–Oppenheimer molecular dynamic simulations.

Often when studying the dynamics of an A + BC system, one begins by constructing a global PES from piecewise additive potential terms. Such semiempirical potentials contain terms that may be systematically varied so as to produce surface features that are reflective of experimental results. This approximation of the potential will work well when the analytic potential is in close agreement with the *ab initio* PES; but, one cannot know if the two are in agreement without the prior calculation of the *ab initio* PES.

If, instead, one wished to simulate the dynamics on the *ab initio* surface, then one is again immediately met with difficulty. First, care is needed in the way one computes the *ab initio* surface, so as to include the needed correlation with higher excited states. Once the list of *ab initio* energies (perhaps even gradients and hessians or local frequencies) that are representative of all energetically accessible asymptotes, as given by the initial conditions, is collected, then typically this list of energy data is fit to an underlying global analytic function (GAF). This requires the varying and fitting of many parameters, and errors become difficult to control and minimize. The *ab initio* surface is free to assume any shape, as marked out by the solution of the time-independent Schrödinger equation, which can make this surface extremely difficult to fit to a GAF. Usually, many parameters are needed so that the function has enough flexibility to represent the *ab initio* data.

An alternative approach is a local methodology that uses *ab initio* data ''near'' a geometry to generate the energy and gradients at that geometry. A local method should have the following attributes: 1) involve propagation on the actual *ab initio* energy surface and not a GAF; 2) be robust and completely general; 3) be capable of classical and quantum simulations; and 4) be computationally tractable. A satisfying way of accomplishing this has been developed in our work and forms the emphasis of this paper. This new technique has been coded in a program called UTIMP-Pack (Utah Tessellation Interpolation Multidimensional Propagation Package). Our method is, in principle, similar to the 2D spline procedure developed by Raff et al.[2,3] and the iterative interpolation techniques of Collins et al.,[4–6] but it differs significantly in practice.

The outline of the article is as follows: The next section introduces and defines a local procedure in two dimensions (2D) and explains the parts that make up our local, 2D methodology. The third section introduces and defines a global, 2D technique and we discuss how this can be facilitated by our work. The fourth section tests both our local and global, 2D methods. We conclude by discussing what remains to be done to facilitate the simulation of any gas-phase A + BC reaction.

Local Methodology

A rigorous mathematical definition of local is a technique, for the generation of function (i.e., PES) or derivative (i.e., gradients) data at a point P, that only requires data points near P to be known *a priori*.[7] In other words, changes in the prior data set from points that are far from the point P will not affect the generation of energy and gradient data at P. Although we use this notion of local in our procedure, we alter this definition to an approach of generating energy and gradient information that is tessellation dependent—dependent upon the domain of the PES being divided into small, localized, and independent pieces.

TESSELLATION

To satisfy the goal of a local procedure, the domain of the PES is divided into simplices; in 2D these are triangles. To be able to divide a surface into simplices barycentric coordinates and their geometric interpretation prove to be useful.[8] In 2D, if one is given three nodes ($p_i = (x_i, y_i)$), that form a simplex, one may find the barycentric coordinates (b_i) of a fourth point ($q = (x_q, y_q)$) relative to the p_is through the solution of eq. (1):

$$\begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1 \\ x_q \\ y_q \end{bmatrix} \quad (1)$$

The x_is and y_is are the pairs of coordinates that make up the domain of the 2D surface. Each barycentric coordinate (b_i) tells the position of q relative to the hyperplane, H_i, formed by the nodes $p_{j \neq i}$. In the 2D case, H_1 is the line connecting p_2 and p_3; H_2 is the line connecting p_1 and p_3; and H_3 will be the line connecting p_1 and p_2. If $b_s = 0$

then q lies in H_s; for $b_s < 0$, q on the opposite side of H_s from p_s; for $b_s > 0$ then q is on the same side of H_s as p_s; see Figure 1.

Because a given domain of a 2D PES will admit to many different triangulations, there must be some method of distinguishing between competing triangulations and finding the optimum one. The optimum triangulation will minimize the number of simplices with one very small internal angle. This optimum triangulation may be facilitated by the so-called sphere test.[8] Consider when one is given four nodes of a convex quadrilateral (all four inner angles formed by the four nodes are less than 180°); see Figure 1. One first chooses any three of the four nodes and finds the circle that lies on these three points. Next, the fourth node is checked to see if it lies inside, outside or on the circle circumscribing the three chosen nodes. If the fourth node lies inside of the circle (as seen in scenario A of Fig. 1), then insert the diagonal from the fourth to the first node. If the fourth node lies outside of the circle (as seen in scenario B of Fig. 1), then insert the diagonal from the second to the third node. If the fourth node lies on the circle, then insert either diagonal.

With these two tools it is possible, in principle, to tessellate an N-dimensional surface. Figure 2 and Table I show the tessellation and output, respectively, for eight scattered points in 2D.

INTERPOLATION

Once the domain has been tessellated, one must be able to interpolate the energy and gradients anywhere on the surface. In the methods that follow, we employ barycentric coordinates [b_i, previously introduced in eq. (1)] because they greatly facilitate the construction and implementation of the interpolant. Two commonly used interpolants given consideration in this work are the Clough–Tocher[9] and the cubic hermite[10] interpolating polynomials. Both are cubic polynomial interpolants and are continuous over the domain (i.e., provide C^0 continuity); however, while the

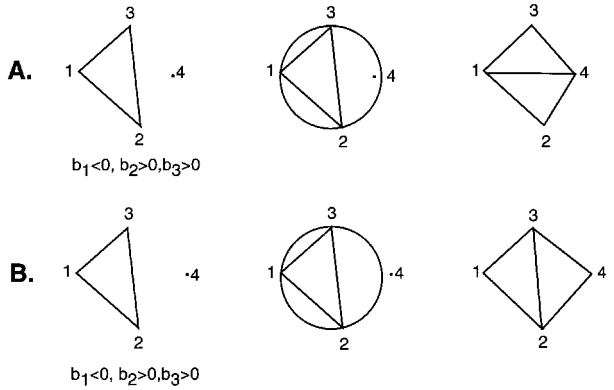

FIGURE 1. Barycentric coordinates of node 4 relative to nodes 1–3. The sphere test applied to nodes 1–4, resulting in the insertion of the diagonal from node 1 to node 4 for senario A and the insertion of the diagonal from node 2 to node 3 for senario B.

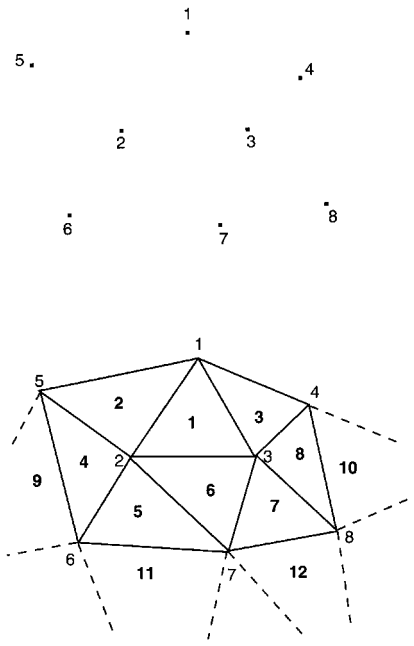

FIGURE 2. The tessellation of nodes 1–8. The bold integers give the triangle index, and the scripted integers give the node index.

TABLE I.
Output for Tessellation of Points of Figure 1.[1]

Δ	Q_1	Q_2	Q_3	T_1	T_2	T_3
1	1	2	3	6	3	2
2	1	5	2	4	1	0
3	3	4	1	0	1	8
4	6	2	5	2	9	5
5	6	7	2	6	4	11
6	7	3	2	1	5	7
7	7	8	3	8	6	12
8	3	8	4	10	3	7
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

[a] Δ is the triangle index, the Q_i's are the node indices and the T_i's ar the bordering triangle indices.

Clough–Tocher is also continuous in the first derivative (C^1 continuity) the cubic hermite is not. It should be noted that both the CT and CH interpolants need energy and gradient values at the nodes of the simplex. Another interpolant that we have tested is a quintic polynomial[11] of the barycentric coordinates. This interpolant requires energy, gradients and hessians at the nodes of the tessellation and, like the CT, is continuous in the energy and gradients (C^1 continuity).

Clough–Tocher (CT)

For the Clough–Tocher interpolant, we begin by considering a triangle in the domain defined by the nodes V_i, where $i = 1, 2, 3$, as displayed in Figure 3. At each node the energy values and the gradients are required as indicated in this figure. However, because we assume a minimal amount of information *a priori*, then at nodes where gradients are not given initially, we can generate them using the methods discussed later. The triangle defined by the nodes V_i is called the macrotriangle which is divided about its centroid [$x_4 = (x_1 + x_2 + x_3)/3$, $y_4 = (y_1 + y_2 + y_3)/3$] into three microtriangles. This is shown in Figure 3, where the microtriangles are labeled by T_i, $i = 1, 2, 3$.

The CT interpolant is displayed in its compact form as

$$E(P) = \sum_{i+j+k+l=3} \frac{3!}{i!j!k!l!} c_{ijkl} b_1^i b_2^j b_3^k b_4^l \quad (2)$$

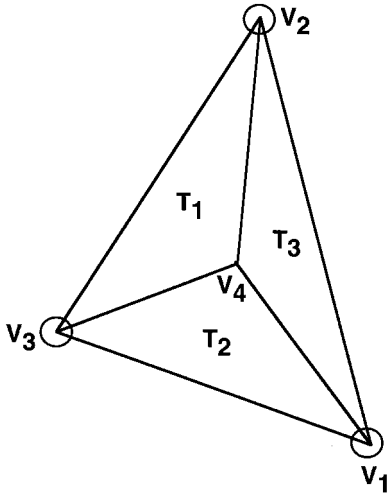

FIGURE 3. Arbitrary macrotriangle showing the microtriangles of which the CT interpolant is dependent. The circle on each node is representative of the nodal gradients.

where the c_{ijkl} are referred to as ordinates and are given in ref. 9 in terms of the values of the energy and the gradient at each V_i. Since gradients give the new force and thus the information needed to carry out dynamics on a potential energy surface, they must be evaluated. The gradient in any direction **e** centered at the point P contained in the macrotriangle is obtained by taking the directional derivative of eq. (2). Taking the directional derivative of eq. (2) gives:

$$\frac{\partial E(P)}{\partial \mathbf{e}} = \sum_{i+j+k+l=2} \frac{2!}{i!j!k!1!} \hat{c}_{ijkl} b_1^i b_2^j b_3^k b_4^l \quad (3)$$

with the ordinates given by:

$$\hat{c}_{ijkl} = 3\left(c_{i+1,j,k,l} \frac{\partial b_1}{\partial \mathbf{e}} + c_{i,j+1,k,l} \frac{\partial b_2}{\partial \mathbf{e}} + c_{i,j,k+1,l} \frac{\partial b_3}{\partial \mathbf{e}} + c_{i,j,k,l+1} \frac{\partial b_4}{\partial \mathbf{e}} \right)$$

where **e** is a vector in $\mathscr{R}^2$.[10] The directional derivative of barycentric coordinates is facilitated by the expression:

$$\frac{\partial b_k}{\partial e_{ij}} = \delta_{kj} - \delta_{ki}$$

where $e_{ij} = V_j - V_i$.

By extending the three vertex data to include the centroid x_4, y_4, the CT interpolant for a macrotriangle is really comprised of three separate interpolants, one for each microtriangle.

To elaborate on this further, consider a point $P(x_P, y_P)$ in the macrotriangle, which must also be a member of one of the microtriangles, say T_2. Since P is located in the microtriangle T_2, we begin by constructing the barycentric coordinates only for the nodes V_1, V_3, and the centroid V_4. The barycentric coordinates corresponding to these three nodes are found by solving the matrix equation:

$$\begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_3 & x_4 \\ y_1 & y_3 & y_4 \end{bmatrix} \begin{bmatrix} b_1 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 1 \\ x_p \\ y_p \end{bmatrix}$$

Once the barycentric coordinates for the microtriangle have been obtained, the barycentric coordinate corresponding to the node V_2 (b_2) is set equal to zero. Using the convention, $0° = 1$, we now have, from eq. (2), an interpolant that gives energy

values on the microtriangle T_2

$$E(P) = \sum_{i+k+l=3} \frac{3!}{i!k!l!} c_{i0kl} b_1^i b_3^k b_4^l$$

Obtaining an interpolant for the gradient at any point in the microtriangle is done in precisely the same manner as interpolating the energy presented above, except now we employ eq. (3). The expression for the gradient in the direction **e** at the point P is given by

$$\frac{\partial E(P)}{\partial \mathbf{e}} = \sum_{i+k+l=2} \frac{2!}{i!k!l!} \hat{c}_{i0kl} b_1^i b_3^k b_4^l$$

and the ordinates are constructed using:

$$\hat{c}_{i0kl} = 3\left(c_{i+1,0,k,l} \frac{\partial b_1}{\partial \mathbf{e}} + c_{i,0,k+1,l} \frac{\partial b_3}{\partial \mathbf{e}} + c_{i,0,k,l+1} \frac{\partial b_4}{\partial \mathbf{e}} \right)$$

In the case where P is in T_3 the barycentric coordinate for b_3 is set equal to zero and the barycentric coordinates for the remaining three nodes along with the interpolants for the energy values and gradients are found as described above. Moreover, if P is an element of T_1, b_1 is set to zero and the same procedure is followed.

Cubic Hermite (CH)

The cubic hermite interpolant is also given in terms of barycentric coordinates which are determined by solving eq. (1). The interpolant is given by:

$$E(P) = \sum_{r=1}^{3} \left[E_r q_r + s_r \left(\frac{\partial E}{\partial x} \right)_r + t_r \left(\frac{\partial E}{\partial y} \right)_r \right] \quad (4)$$

where:

$$E_r, \left(\frac{\partial E}{\partial x} \right)_r, \text{ and } \left(\frac{\partial E}{\partial y} \right)_r$$

are the energies and gradients at the nodes of the simplex, respectively. The barycentric coordinates appear in the expressions:

$$q_i = b_i^2(3 - 2b_i) + 2b_i b_j b_k$$

$$s_i = b_i^2(\gamma_k b_j - \gamma_j b_k) + \frac{1}{2} b_i b_j b_k (\gamma_k - \gamma_j)$$

$$t_i = b_i^2(\beta_j b_k - \beta_k b_j) + \frac{1}{2} b_i b_j b_k (\beta_j - \beta_k)$$

where:

$$\beta_i = y_i - y_k \text{ and } \gamma_i = -(x_j - x_k)$$

and (i, j, k) is a cyclic permutation of (1, 2, 3). Obtaining the gradient can be done by differentiating eq. (4). Because the only dependence on x and y in eq. (4) is located in the barycentric coordinates, the gradient in any direction **e** in the domain is given by

$$\frac{\partial E(P)}{\partial \mathbf{e}} = \sum_{r=1}^{3} \left[E_r \left(\frac{\partial q}{\partial \mathbf{e}} \right)_r + \left(\frac{\partial s}{\partial \mathbf{e}} \right)_r \left(\frac{\partial E}{\partial x} \right)_r + \left(\frac{\partial t}{\partial \mathbf{e}} \right)_r \left(\frac{\partial E}{\partial y} \right)_r \right]$$

Although the CH interpolant is not C^1 continuous, it does not require the splitting of the macrotriangle into three microtriangles as is needed for the CT.

Quintic Polynomial (Q18)

Another interpolant considered is the 18 term, fifth-degree polynomial given by:

$$E(P) = \sum_{r+s+t=5} \frac{5!}{r!s!t!} c_{rst} b_1^r b_2^s b_3^t \quad (5)$$

The Q18 is similar to both the CT and CH because it can also be expressed in the more tractable form using barycentric coordinates. Like the CH, the barycentric coordinates are found by solving eq. (1) that is, it is not microtriangle-dependent. The ordinates of the Q18 interpolant are like those used in the CT in that they depend upon energy and gradients at the nodes of the simplex; however, for the Q18, additional information in the form of hessians at the nodes is needed. The ordinates for the Q18 interpolant may be found in ref. 11.

Gradients for the Q18 interpolant are given by the expression:

$$\frac{\partial E(P)}{\partial \mathbf{e}} = \sum_{r+s+t=4} \frac{4!}{r!s!t!} \hat{c}_{rst} b_1^r b_2^s b_3^t$$

where the ordinates are:

$$\hat{c}_{rst} = 3\left(c_{r+1,s,t} \frac{\partial b_1}{\partial \mathbf{e}} + c_{r,s+1,t} \frac{\partial b_2}{\partial \mathbf{e}} + c_{r,s,t+1} \frac{\partial b_3}{\partial \mathbf{e}} \right)$$

GRADIENT GENERATION

Methods 1 and 2 from the previous section require knowledge of the energy and gradients, whereas method 3 requires knowledge of the energy, gradients, and hessians at all nodes. Because the computation of *ab initio* gradients and hessians are expensive, and thus are assumed to rarely be part of the data set, UTIMP-Pack has the capability of generating them. The methods of gradient and hessian generation discussed in this study are the hyperbolic multiquadric,[12,13] the weighted least squares,[13] and the thin plate spline.[11] A brief discussion of each method and the testing results are given below.

Hyperbolic Multiquadric (MUL)

The hyperbolic multiquadric function:

$$E_j(x_j, y_j) = \sum_{i=1}^{K} c_i \sqrt{d_{ji}^2(x_{j,i}, y_{j,i}) + r} \qquad (6)$$

where

$$d_{ji}^2(x_{j,i}, y_{j,i}) = (x_j - x_i)^2 + (y_j - y_i)^2$$

can be used as both a local and global interpolant, in the rigorous mathematical sense. (For the global definition see Global Methodology section.) The multiquadric interpolant has two *ad hoc* parameters r and K, that are correlated to the grid spacing. The interpolant is quite stable with changes in both parameters.[7]

When used in the rigorous mathematical notion of local, one chooses, inclusively, the K closest points to the node where the gradients are sought. Once the c_i coefficients are found, through the solution of the $K \times K$ matrix equation:

$$\begin{bmatrix} d_{11} & \cdot & \cdot & d_{1k} \\ \cdot & \cdot & & \cdot \\ \cdot & & \cdot & \cdot \\ d_{kl} & & & d_{kk} \end{bmatrix} \begin{bmatrix} c_1 \\ \cdot \\ \cdot \\ c_k \end{bmatrix} = \begin{bmatrix} E_1 \\ \cdot \\ \cdot \\ E_k \end{bmatrix}$$

then eq. (6) may be differentiated to yield the gradients at the node of interest.

The method of minimizing the first difference in the gradients is used to find the optimum r parameter. This method initializes r to zero and computes the gradients. Then r is gradually increased and the gradients are computed with each new r. The optimum r is the r that minimizes the first difference in the gradients.

Weighted Cubic Least Squares (LS)

Another technique to generate the gradient at the nodes of a triangle is the weighted cubic least squares. The cubic polynomial:

$$\begin{aligned} E(x, y; x_i, y_i; \beta_1 \ldots \beta_9) &= E_0 + \beta_1(x - x_i) \\ &\quad + \beta_2(y - y_i) + \beta_3(x - x_i)^2 \\ &\quad + \beta_4(x - x_i)(y - y_i) \\ &\quad + \beta_5(y - y_i)^2 + \beta_6(x - x_i)^3 \\ &\quad + \beta_7(x - x_i)^2(y - y_i) \\ &\quad + \beta_8(x - x_i)(y - y_i)^2 + \beta_9(y - y_i)^3 \end{aligned}$$

is used to approximate the function in a neighborhood of the point (x, y). This is done by varying the coefficients $\beta_1, \ldots, \beta_9$ to minimize the quantity:

$$\chi^2 = \sum_{i=1}^{K} w_i(x, y)[E_i - E(x, y; x_i, y_i; \beta_1 \ldots \beta_9)]^2$$

where, once again, K is the number of closest points to (x, y). Minimizing χ^2 with respect to the coefficients β_i is done by solving the system of nine equations, linear in the coefficients, resulting from:

$$\frac{\partial E}{\partial \beta_i} = 0$$

where $i = 1, \ldots, 9$.

The weights $w_i(x, y)$ tested for this method were:

1. minimizing the first difference in the gradients (see Hyperbolic Multiquadric subsection) (LS1)
2. $(R - d_i(x, y))^2$ (LS2)
3. $\exp[-(R - d_i(x, y))^2]$ (LS3)
4. 1 (LS4)

where R is the distance from (x, y) to the farthest of the K closest points.

Thin Plate Spline (TPS)

The thin plate spline method of gradient generation makes use of the variational principle to minimize the functional[8]

$$F = \sum_{(T)} \phi$$

where:

$$\phi = \int\int_T \left(\left| \frac{\partial^2 E}{\partial x^2} \right|^2 + 2 \left| \frac{\partial^2 E}{\partial x \partial y} \right|^2 + \left| \frac{\partial^2 E}{\partial y^2} \right|^2 \right) dxdy$$

E is as given in Eq. 5. ϕ involves the integration over a single triangle, T, in the tessellation.

The nodal energies, gradients, and hessians are nested in the ordinates of eq. (5). The nodal gradients and hessians are the parameters that are used to bring about the minimization of the functional (F). The thin plate spline may be thought of as clamping a thin, elastic plate through all the energy points of a PES and using the gradients and the hessian as the parameters to bring about the minimization of the stiffness of this plate.

This method has a mathematical basis and as such does not suffer from parameters that are *ad hoc*, as with the MUL and LS methods. Because of this mathematical basis, the method is global in the rigorous mathematical sense. However, the method requires the tessellation of the domain and is, therefore, local as we have defined it for the purposes of this article. The method suffers, however, from necessitating the solution of a $5N \times 5N$ sparse system of linear equations, where N is the total number of data points known *a priori*.

INTEGRATOR FOR EQUATIONS OF MOTION

The Adams fourth-order predictor-corrector algorithm is the integrator used to evaluate the first-order equations of motion. To initiate propagation, the first four starting values of the trajectory are obtained using the fourth-order Runge–Kutta method. Once the four starting values are generated, propagation is then turned over to the Adams–Bashforth four-step method as predictor to generate the next value in the propagation where this value is then corrected using the Adams–Moulton three-step method. Propagation continues from this point using only the Adams–Bashforth method as predictor followed by the Adams–Moulton method as corrector.[14]

Global Methodology

A rigorous mathematical definition of a global methodology for the generation of energy and gradient information at a point P is a method that utilizes all the data points given *a priori*.[7] Therefore, changes in a prior data point alters the generation of energy and gradients at all points. Once again, for the purposes of this study, we have altered this definition so that a global methodology is one that does not necessitate the tessellation of the surface. In other words, this methodology uses the list of *ab initio* energies solely to run the dynamics of the system.

HYPERBOLIC MULTIQUADRIC (MUL-G)

The hyperbolic multiquadric was introduced earlier in eq. 6. In the second section, this interpolant was introduced in the local regime, where the K closest points to a node were used to generate the gradients at the node. If, instead, we set K equal to the total number of data points given *a priori*, then the local technique becomes a global interpolant. This means that anywhere on the surface we may generate the energy and gradients after the initial $K \times K$ matrix is solved. This matrix only needs to be solved once. After the c_is of eq. 6 are found, then the gradients at each time step may be computed by differentiating this equation. Once again, this technique makes the tessellation of the surface unnecessary, thereby doing away with the computational costs of tessellating. However, this asset is quickly eroded as the computational cost of each trajectory increases. For each gradient calculation a sum over the total number of data points is needed.

In the global regime, the r parameter becomes a bit more difficult to choose. One may exclude an arbitrary number of nodes from the data set, find the coefficients using the remaining nodes, and calculate the energy at the excluded points. Then r is gradually advanced and the procedure is repeated, until the optimum r is found by the minimization of the absolute error in the energy. Alternatively, one may select arbitrary geometries that are not contained in the data set used to find the MUL coefficients and apply the method of minimizing the first difference in the energy and/or gradients that was introduced earlier in this article.

Results

It is worthy of mention that all units are expressed in atomic units and that, for all data sets, the input coordinates and potential energy contained accuracy up to the millionth place. We felt

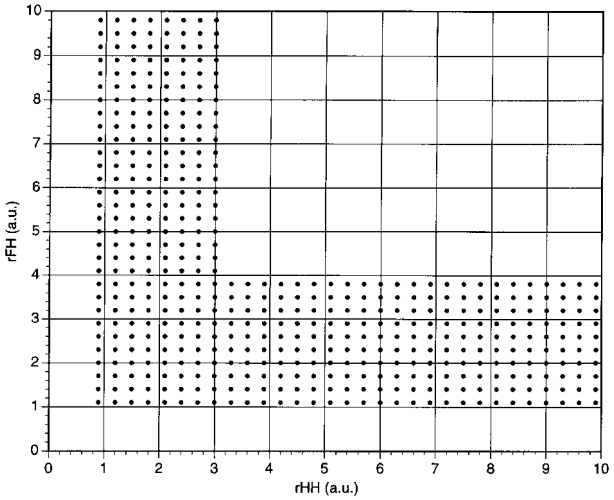

FIGURE 4. The regularly space 470 node domain for DS1.

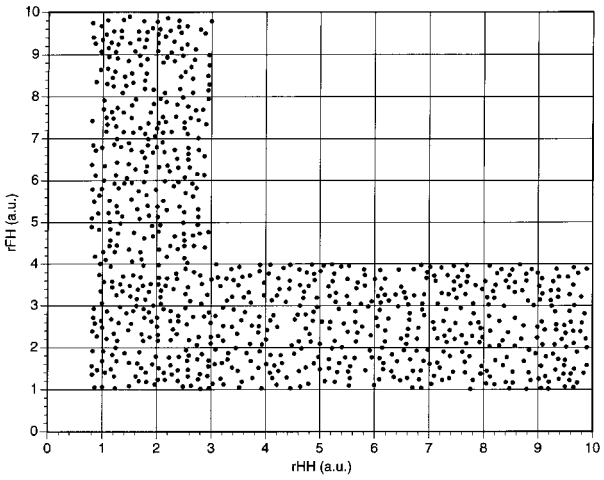

FIGURE 6. The randomly scattered 800-node domain for DS2.

this accuracy would be representative of the output of many standard quantum chemistry packages.

To validate our methodology, tests were performed on the 2D, collinear $F + H_2$ semiemperical potential energy surface developed by Muckerman.[15] We have generated two different data sets with potential energy data only known *a priori*. Data set 1 (DS1, Figs. 4 and 5) is a regularly spaced set of 470 points (at a 0.3-a.u. spacing) covering the H_2 and FH zero point regions. Data set two (DS2, Figs. 6 and 7) is a randomly scattered set of 800 points also covering the H_2 and FH zero point regions. The lack of smoothness in the contours of DS2 is due to the 3D graphic software employed and has nothing to do with the methodology discussed in this article.

We have used the methods of gradient generation mentioned earlier and examined the errors of each in generating nodal gradients for DS1 and DS2. These nodal gradients were then utilized in the interpolants to generate the potential and gradients at the centroids of the tessellation and compared with the analytic results. In addition, the continuity of the energy and gradients across the boundary of the triangles was also examined. Lastly, two trajectories, from the local and global regimes, were executed and the continuity of the energy and angular momentum examined.

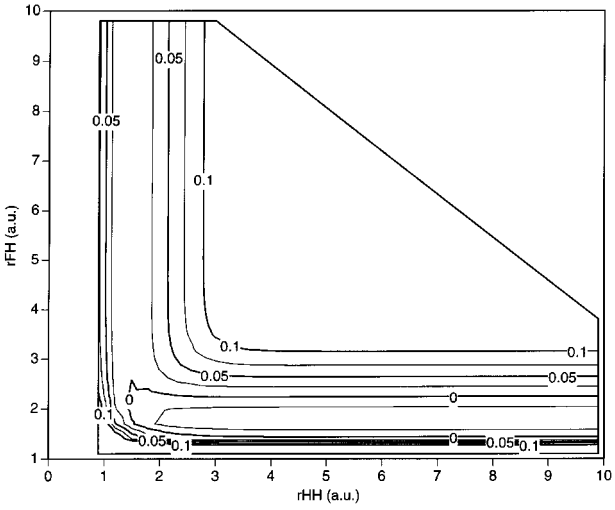

FIGURE 5. The $F + H_2$ PES for DS1.

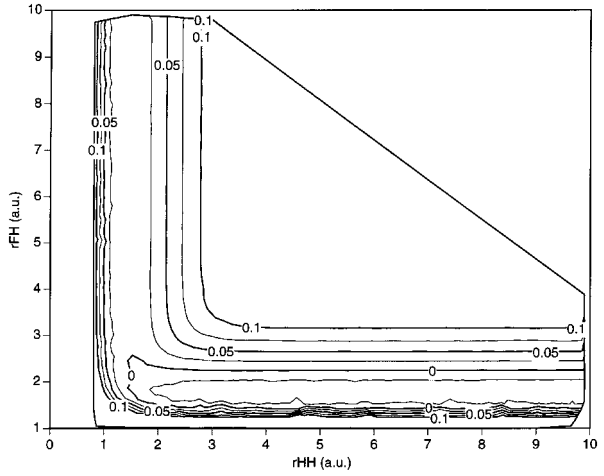

FIGURE 7. The $F + H_2$ PES for DS2.

COMPARISON OF GRADIENT-GENERATING TECHNIQUES

There is one *ad hoc* parameter at our disposal for both the MUL and LS methods—K the number of surrounding data points used in the generation of the gradients at any particular node. We examined a number of K values for both data sets and selected the data for $K = 49$ to report. Tables II and Table III show the errors in the nodal gradients using the MUL, LS, TPS, and MUL-G methods on DS1 and DS2, respectively.

The local methods (LS1, LS2, LS3, LS4, MUL, and TPS) give at least one place of accuracy in the average absolute errors in the gradients on DS1, as shown in Table II. The MUL method is superior to all other local methods by its ability to maintain two places of accuracy in the average absolute errors in the gradients while having maximum absolute errors that are lower than the other local methods by a factor of 10. Comparing the results of MUL and the MUL-G methods shows that using the MUL method in the local regime does not severely hinder its gradient generation ability for DS1, as the average and maximum errors in the gradients are on par with one another.

Examining Table III, which shows the errors in the gradient-generation techniques for DS2, again reveals that the MUL method is superior to other local methods by generating average and maximum absolute errors that are lower by a factor of 10. Comparing the MUL and MUL-G methods shows that the global approach is better by giving smaller errors about twice as often as the local approach.

Comparing the LS and MUL methods on DS1 and DS2 for K values equal to 9, 13, 21, 25, and 37, we found that the LS and MUL methods usually gave at least two digits of accuracy in the average absolute gradient errors. With at least 21 and 13 points used in the generations of the gradients for DS1 and DS2, respectively, both the LS and the MUL methods showed stability with an increasing K value.

Comparing Tables II and III shows that, in the local regime, the MUL method is the most robust as indicated in its ability to maintain at least two digits of accuracy in the average absolute errors in the gradients, while maintaining one digit of accuracy in the maximum absolute errors in the gradients for DS1 and DS2. Consistent in both tables, and demonstrated particularly well in Table III, was the MUL-G method giving superior results when compared with the local methods. The LS methods give better results with randomly spaced data than with regularly space data, as indicated by their ability to maintain both lower average and lower maximum absolute errors in the gradients. The TPS method gives results similar to the LS methods. Because DS2 contains 330 more data points over the same region as DS1, we see that most methods pick up one place of accuracy in the average absolute errors in the gradients, which must be due to the closer packing of the 49 points.

In summary, Tables II and III demonstrate that the MUL gradient-generation method, when used in both the local and global regimes, is the most robust and has fewer errors than the local LS and TPS methods.

TABLE II.
Comparison of Methods of Gradient Generation for DS1 Using LS and MUL Methods with 49 Surrounding Data Points and TPS and MUL-G Methods.

	a	*b*	*c*	*d*	*e*	*f*
LS1	108	55	.006559	.020486	.149079	.333057
LS2	71	48	.005879	.017272	.111139	.256929
LS3	7	4	.011730	.031229	.171059	.377404
LS4	4	2	.010583	.028728	.160116	.359511
MUL	152	98	.001214	.005075	.011208	.036406
TPS	33	31	.011228	.032529	.155584	.377942
MUL-G	95	232	.001169	.001407	.016113	.020569

a = number of times that the method gave the smallest absolute error in the *rHH* gradient; *b* = number of times that the method gave the smallest absolute error in the *rFH* gradient; *c* = average absolute error in the *rHH* gradient; *d* = average absolute error in the *rFH* gradient; *e* = maximum absolute error in the *rHH* gradient; *f* = maximum absolute error in the *rFH* gradient.

TABLE III.
Comparison of Methods of Gradient Generation for DS2 Using LS and MUL Methods with 49 Surrounding Data Points and TPS and MUL-G Methods.

	a	*b*	*c*	*d*	*e*	*f*
LS1	32	29	.003110	.007494	.132033	.234049
LS2	30	11	.002801	.007246	.089986	.200930
LS3	8	6	.005444	.013440	.148145	.329142
LS4	9	3	.004981	.012520	.137429	.292731
MUL	214	217	.000296	.000412	.013604	.009667
TPS	19	25	.006602	.014810	.142277	.414890
MUL-G	488	509	.000049	.000106	.002777	.009844

a = number of times that the method gave the smallest absolute error in the *rHH* gradient; *b* = number of times that the method gave the smallest absolute error in the *rFH* gradient; *c* = average absolute error in the *rHH* gradient; *d* = average absolute error in the *rFH* gradient; *e* = maximum absolute error in the *rHH* gradient; *f* = maximum absolute error in the *rFH* gradient.

COMPARISON OF CT, CH, Q18, AND THE MUL-G INTERPOLANTS

Employing the gradients generated by the MUL method using 49 surrounding data points in the CT and CH interpolants, we may compare these two interpolants alone and with the Q18 and the MUL-G methods in their ability to generate the energy and gradients at the centroids, and in their continuity of the energy and gradients across the boundaries of the triangulation.

Tables IV and V show that for DS1 and DS2, respectively, using the analytic energies, gradients and hessians, the Q18 is far superior to the CT and CH methods. However, when the gradients and hessians generated from the TPS are used and compared with the CT and CH using the gradients from the MUL method, all methods give at least two places of accuracy in the average absolute gradient errors and it appears that the CH method may have a slight advantage in its ability to maintain both lower average and lower maximum absolute errors in the gradients at the centroid. Consistent in both tables is that the MUL-G method is superior to all local approaches where gradients are generated by the MUL method.

Comparing Tables IV and V shows that, on average, both the CT and CH interpolants have the ability to maintain three digits of accuracy in the potential and two digits of accuracy in the gradients at the centroid while using nonanalytic gradients. The most pronounced effect of the data sets is found in the maximum absolute errors, where, for DS2, much larger maximum absolute errors in gradients at the centroid are found. This is probably a result of an edge triangle with a very small angle in the tessellation. In general, interior trian-

TABLE IV.
Comparison of Methods for Generation of Potential and Gradients at Centroids of Tessellation for DS1.

	a	*b*	*c*	*d*	*e*	*f*
CT[a]	.000111	.001464	.001580	.001261	.014548	.017463
CH[a]	.000205	.000132	.000403	.002179	.001527	.003517
Q18[b]	.000002	.000016	.000017	.000015	.000155	.0000159
CT[c]	.000312	.002542	.002816	.004270	.043419	.052998
CH[c]	.000338	.001131	.001893	.004556	.023732	.039120
Q18[d]	.001190	.002708	.006797	.018006	.045018	.108871
MUL-G	.001037	.000528	.000295	.003104	.003561	.005389

a = average absolute error in the potential; *b* = average absolute error in the *rHH* gradient; *c* = average absolute error in the *rFH* gradient; *d* = maximum absolute error in the potential; *e* = maximum absolute error in the *rHH* gradient; *f* = maximum absolute error in the *rFH* gradient.
[a] Analytic gradients; [b] analytic gradients and hessians; [c] gradients from the MUL with 49 surrounding points; [d] gradients and hessians from the TPS.

TABLE V.
Comparison of Methods for Generation of Potential and Gradients at Centroids of Tessellation for DS2.

	a	*b*	*c*	*d*	*e*	*f*
CT[a]	.000061	.001857	.001080	.002769	.800283	.056977
CH[a]	.000084	.000976	.000314	.003579	.529172	.026855
Q18[b]	.000001	.000150	.000008	.000620	.118250	.000674
CT[c]	.000076	.002081	.001252	.002835	.799033	.051578
CH[c]	.000098	.001232	.000534	.004730	.528278	.023371
Q18[d]	.000450	.003175	.003192	.020854	.192516	.121342
MUL-G	.000003	.000020	.000018	.000158	.002947	.001298

a = average absolute error in the potential; *b* = average absolute error in the *rHH* gradient; *c* = average absolute error in the *rFH* gradient; *d* = maximum absolute error in the potential; *e* = maximum absolute error in the *rHH* gradient; *f* = maximum absolute error in the *rFH* gradient.
[a] Analytic gradients; [b] analytic gradients and hessians; [c] gradients from the MUL with 49 surrounding points; [d] gradients and hessians from the TPS.

gles give superior results than do the exterior triangles, which is probably due to the lower errors in the interior nodal gradients. When comparing the local methods to the MUL-G method, it is noticed that one and two places of accuracy are picked up in the average absolute gradient errors in Tables IV and V, respectively.

When using these interpolants for the propagation of a particle or wavepacket it is very important to have continuity in the energy and gradients across the boundaries of the triangles. Tables VI and VII examine the continuity of each interpolant by taking the midpoints of the three sides of the triangles of the tessellation, moving both forward and backward along the vector perpendicular to the triangle sides by a distance ε and calculating the energy and gradients. Moving along this perpendicular will place the two points, where the energy and gradients are computed, a short distance from one another and in two adjacent triangles.

Examining Tables VI and VII for DS1 and DS2, respectively, shows that, as expected, the CT and Q18 interpolants demonstrate C^1 continuity (continuity of the potential and the gradients). The CH interpolant, which is C^0 continuous, shows this by the continuity in the potential and the discontinuity in the fourth and third decimal place for the gradients of DS1 and DS2, respectively, when using non-analytic gradients at the nodes. Continuity for DS2 is much more difficult for all interpolants, as shown by the maximum absolute differences in

TABLE VI.
Continuity of Potential and Gradients by Two Points, in Differing Triangles, Along the Perpendicular Centered at the Midpoints of the Sides of the Triangles of the Tessellation for DS1.

	a	*b*	*c*	*d*	*e*	*f*
CT[a]	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	.000007	.000009	.000027
CH[a]	$< 10^{-6}$	.000002	.000650	.000006	.000644	.016432
Q18[b]	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	.000006	.000009	.000031
CT[c]	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	.000007	.000009	.000029
CH[c]	$< 10^{-6}$	.000202	.000931	$< 10^{-6}$	.013572	.031130
Q18[d]	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	.000002

These two points are separated by $3.2 * 10^{-6}$ au. *a* = average absolute difference in the potential; *b* = average absolute difference in the *rHH* gradient; *c* = average absolute difference in the *rFH* gradient; *d* = maximum absolute difference in the potential; *e* = maximum absolute difference in the *rHH* gradient; *f* = maximum absolute difference in the *rFH* gradient.
[a] Analytic gradients; [b] analytic gradients and hessians; [c] gradients from the MUL with 49 surrounding points; [d] gradients and hessians from the TPS.

TABLE VII.
Continuity of Potential and Gradients by Two Points, in Differing Triangles, Along the Perpendicular Centered at the Midpoints of the Sides of the Triangles of the Tessellation for DS2.

	a	*b*	*c*	*d*	*e*	*f*
CT[a]	$< 10^{-6}$	$< 10^{-6}$	.000001	000014	.000257	.000522
CH[a]	$< 10^{-6}$	.002263	.006659	.000014	.534775	2.631390
Q18[b]	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	.000014	.000045	.000226
CT[c]	$< 10^{-6}$	$< 10^{-6}$	.000001	.000014	.000242	.000518
CH[c]	$< 10^{-6}$	.002329	.006681	.000007	.532713	2.621250
Q18[d]	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	.000002	.000004

These two points are separated by $2.5 * 10^{-6}$ au. *a* = average absolute difference in the potential; *b* = average absolute difference in the *rHH* gradient; *c* = average absolute difference in the *rFH* gradient; *d* = maximum absolute difference in the potential; *e* = maximum absolute difference in the *rHH* gradient; *f* = maximum absolute difference in the *rFH* gradient.
[a] Analytic gradients; [b] analytic gradients and hessians; [c] gradients from the MUL with 49 surrounding points; [d] gradients and hessians from the TPS.

the gradients, which is most likely due to the irregularly shaped triangles of a randomly scattered data set.

In summary, among the methods of interpolation of the energy and gradients using nonanalytic gradients, it is apparent that, for DS1 and DS2, the MUL-G method consistently gives the lowest average and maximum absolute errors. Within the local approaches, the CH interpolant is superior in both the average and maximum absolute errors at the centroid, but has the undesirable attribute of C^0 continuity. For this reason, among the local approaches, the CT interpolant is preferred.

TRAJECTORIES ON THE $F + H_2$ SURFACE USING THE LOCAL AND GLOBAL METHODOLOGIES

In this subsection, we note that the trajectories shown were not selected on the basis of their reproducibility. In fact, in the local regime, the CH method was chosen because it demonstrated the largest errors in the continuity of the gradients across the boundaries of triangles on the DS2 surface. This was done to examine how the discontinuity in the gradients effects the continuity of the energy and angular momentum. The second trajectory was facilitated by the MUL-G method on the DS2 surface. The two trajectories shown have the same initial conditions.

As shown in Figure 8, the two trajectories begin to diverge as they approach the transition state region. The CH trajectory calculates gradients along the *rFH* coordinate that are slightly less than zero, whereas the MUL-G method calculates gradients along the *rFH* coordinate that are closer to zero. The nonzero gradients from the CH are most likely a result of the MUL gradients that were generated at the nodes. Table III shows that, for DS2, the MUL gradients had, on average, three places of accuracy. If, however, the analytic gradients in a certain region are very near zero, then the MUL gradients will likely contain errors that are over or underestimated by a factor of ca. 10^3. The *rHH* coordinate for both trajectories stay very close to one another in both amplitude and phase. But, because of the *rFH* coordinate, the two trajectories diverge and produce nonreactive and reactive outcomes for the local CH method and the MUL-G method, respectively.

In Table VII, the CH demonstrated two places of accuracy in the average absolute difference in two gradients in adjacent triangles. As seen in

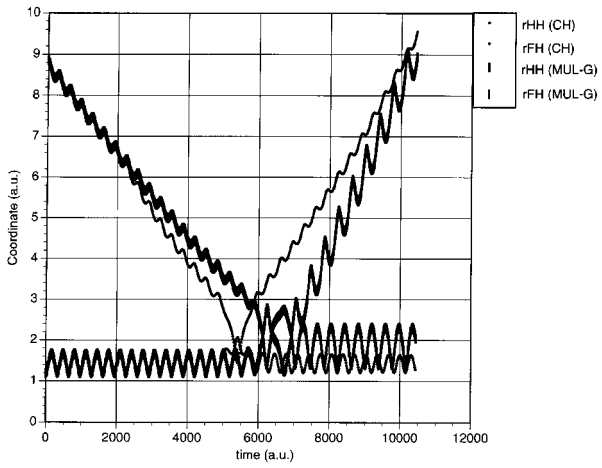

FIGURE 8. Trajectories for the CH interpolant (using gradients generated by the 49-point MUL method) and the MUL-G method on DS2.

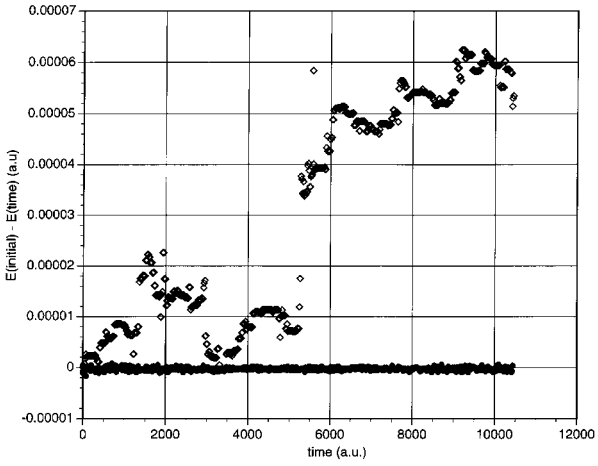

FIGURE 9. Conservation of the energy for the two trajectories shown in Figure 8. The open diamonds and dots are the differences in the total energy from initial value and the value at every 30th time step for local CH and MUL-G trajectories. The change in angular momentum was indistinguishable from zero for both trajectories and is, therefore, not shown.

Figure 9, this average discontinuity in the gradients between triangles manifests a maximum absolute error of ca. $6*10^{-5}$ a.u. in the energy and changes in the angular momentum that are indistinguishable from zero. The MUL-G interpolant (which is tessellation-independent and infinitely differentiable) demonstrates changes of $1*10^{-6}$ in energy, whereas changes in the angular momentum are also indistinguishable from zero.

In summary, the data on DS1 and DS2 shows that the MUL-G approach is superior from the errors in the energy and gradients, but this method is also the most computationally expensive. It is for this reason, that the local approach is the best compromise. Among the local methods, the MUL method of gradient generation combined with the CT interpolant is believed to be the best.

Summary

The methodology that has been described has some features that make going to higher dimensions computationally feasible. It is readily admitted that this method, as well as all others, will suffer from the number of data points needed for propagation scaling to the power of the dimension. But, at least two features make the method described here superior to many others. First, the method allows for a nonuniform density of points. So, in areas where the potential is fairly constant, fewer points need to be used; as, in areas where the potential is undergoing large changes or areas expected to be of critical importance to the outcome of the reaction (i.e., transition state, barriers, etc.), a higher density of points may be used. This attribute may also be utilized in not computing any points in areas that are known to be inaccessible to trajectories due to energy considerations. Second, where direct dynamical methods suffer from the number of data points that need scaling to the dimension of the system on each trajectory computed, this method suffers from this problem with the initial computation of the PES only. Once one has a list of *ab initio* energies, no further *ab initio* calculations are needed. From these two points it is evident that the computational bottleneck in applying this methodology lies in the construction of the input PES, of which the methodology is independent. This computational bottleneck is already surmountable for 3D systems. Once one has a list of *ab initio* energies (perhaps even gradients), then tessellating and running the simulations on this potential should be computationally insignificant relative to the time spent computing the PES. Because of the tractability of this methodology, many ensemble averages of initial states may be simulated, so that statistical errors are small.

In this report we have investigated the reliability of a completely general methodology in 2D. This methodology is in the process of being extended and tested in 3D. The main differences are the number of data points used will be larger, the domain of the PES will be tessellated with tetrahedrons and the interpolants utilized will need to insure continuity across the faces and edges of tetrahedrons. Upon completion of the work in 3D, a completely general methodology—capable of classical and quantum simulation and computationally tractable—will be in place.

Acknowledgments

The authors thank Peter Alfeld for help with the TPS and other questions regarding this project. We also thank Jack Simons for reviewing the manuscript and for discussions that were helpful in the validation of our work.

References

1. E. Deumens, A. Diz, R. Longo, and Y. Öhrn, *Rev. Mod. Phys.*, **66**, 917 (1994).
2. D. R. McLaughlin and D. L. Thompson, *J. Chem. Phys.*, **59**, 4393 (1973).
3. N. Sathyamurthy and L. M. Raff, *J. Chem. Phys.*, **63**, 464 (1975).
4. J. Ischtwan and M. A. Collins, *J. Chem. Phys.*, **100**, 8080 (1994).
5. M. J. T. Jordan, K. C. Thompson, and M. A. Collins, *J. Chem. Phys.*, **102**, 5647 (1995).
6. M. J. T. Jordan, K. C. Thompson, and M. A. Collins, *J. Chem. Phys.*, **103**, 9669 (1995).
7. R. Franke, *Math. Comp.*, **38**, 181 (1982).
8. C. L. Lawson, *Comput.-Aided Geom. Design*, **3**, 231 (1986).
9. P. Alfeld, *Comput.-Aided Geom. Design*, **1**, 169 (1984).
10. M. K. Jain, In *Numerical Solution of Differential Equations, 2nd Ed.*, Halstead Press, New York, 1984, p. 538.
11. P. Alfeld, *Comput.-Aided Geom. Design*, **2**, 281 (1985).
12. R. L. Hardy, *J. Geophys. Res.*, **76**, 1905 (1971).
13. S. E. Stead, *Rocky Mountain J. Math.*, **14**, 265 (1984).
14. R. L. Burden and J. D. Faires, In *Numerical Analysis, 4th Ed.*, PWS-Kent, Boston, 1989, p. 262.
15. J. T. Muckerman, *J. Chem. Phys.*, **54**, 1155 (1971).